

Redis总结

redis获取/设置密码

config get requirepass | config get repass

config set requirepass "new password"

string

set key value

get key

del key

mset key1 value1 key2 value2 ...

mget key1 key2 ...

获取数据字符个数（字符串长度）

strlen key

追加信息到原始信息后部（如果原始信息存在就加，否则新建）

append key value

设置数值数据增加指定范围的值

incr key

incrby key increment

incrbyfloat key increment

设置数值数据减少制定范围的值

decr key

decrby key increment

按数值进行操作的数据，如果原始数据不能转换为数值，或则超过了Redis数值范围上限，将报错。

设置数据具有指定的生命周期

setex key seconds value

psetex key milliseconds value

Redis控制数据的生命周期，通过数据是否失效控制业务行为，适用于所有具有时效性限定控制的操作

常用于存储的格式：

表名:主键名:主键值 **value**

hash

hset key field value

hget key field

hgetall key

hdel key field

hmset key field1 value1, field2 value2,

hmget key field1, field2,

hkeys key

hvals key

hincrby key field increment

hincrbyfloat key field increment

hash类型下的value只能存储字符串，不能再嵌套存储其他类型的数据

每个hash可以存储 $2^{32} - 1$ 个键值对

hash的设计初衷不是为了存储对象的，因此不可滥用

hgetall操作可以获取全部属性，如果内部field过多，效率很低

string存储对象（json）与hash存储对象，string在于存储，hash在于修改

购物车实现

以用户ID作为key，商品编号作为field，商品数量作为value

当前仅仅是将数据存储到了Redis中，并没有起到加速的作用，商品信息还需要二次查询数据库

- 每条购物车中的商品记录保存成两条field

- field1专用于保存购买数量

命名格式：商品ID：nums

保存数据：数值

- field2专用于保存购物车中显示的信息，包含文字描述，图片地址，所属商家信息等

命名格式：商品ID：info

保存数据：json

hmset 001 g01:nums 2 g01:info {.....}

商品信息可以独立hash，这样可以大大减少重复的信息

hsetnx key field value

商家应用

- 以商家ID作为key
- 将参与抢购的商品ID作为field
- 将参与抢购的商品数量作为对应的value
- 抢购时使用降值的方式控制产品的数量

Redis通常只做数据的提供和保存，尽量不要把业务压到Redis上

Tips5:

Redis应用于抢购，限购类、限量发放优惠券、激活码等业务的数据存储设计

list

添加/修改

lpush key value1 [value2]....

rpush key value1 [value2]....

获取数据

lrange key start stop (0, -1显示所有数据)

lindex key index

llen key

获取并移除数据

lpop key

rpop key

规定时间内获取并移除数据

blpop key1 [key2] timeout

brpop key1 [key2] timeout

朋友圈点赞，按照点赞顺序现实好友的点赞信息

如果取消点赞，移除对应好友信息

- 移除指定数据

lrem key count value （value为从左边开始删除指定个数的value）

Tips6:

redis应用于具有操作先后顺序的数据控制

list中保存的数据都是string类型的，数据总量是有限的

list中最多保存 2^{32} - 个元素

list具有索引的概念，但是操作数据时通常以队列的形式进行入队操作，或以栈的形式进行入栈出栈操作

获取全部数据操作结束索引设置为-1

list可以对数据进行分页操作，通常第1页的信息来自于list，第2页及更多的信息通过数据库的形式加载

应用场景

微博/twitter关注列表/粉丝列表，先关注的在前面；

新闻、咨询类网站如何将新闻或咨询按照发生的时间顺序展示；

企业运营过程中，系统将产生大量的运营数据，如何保障多台服务器操作日志的统一顺序输出？（每个服务器都往Redis中写）

解决方案

- 依赖list的数据具有顺序的特征对信息进行管理
- 使用队列模型解决多路信息汇总合并的问题
- 使用栈模型解决最新消息的问题

set

添加数据

sadd key member1 [member2]

获取全部数据

smembers key

删除数据

srem key member1 [member2]

获取集合数据总量

scard key

判断集合中是否包含指定数据

sismember key member

Set类型数据的扩展操作

随机获取集合中指定数量的数据

randmember key [count]

随机获取集合中的某个数据并将该数据移除集合

spop key

求两个集合的交、并、差集

sinter key1 [key2]

sunion key1 [key2]

sdiff key1 [key2]

求两个集合的交、并、差集并存储到指定集合中

sinterstore destination key1 [key2]

sunionstore destination key1 [key2]

sdiffstore destination key1 [key2]

将指定数据从原始集合中移动到目标集合中

smove source destination member

sorted_set

在set的存储结构基础上添加可排序字段

添加数据

zadd key score1 member1 [score2 member2]

获取全部数据

zrange key start stop [WITHSCORES] (0 -1现实所有的数据)

zrevrange key start stop [WITHSCORES]

删除数据

zrem key member [member ...]

按条件获取数据

zrangebyscore key min max [WITHSCORES] [LIMIT]

zrevrangebyscore key max min [WITHSCORES]

条件删除数据

zremrangebyrank key start stop

zremrangebyscore key min max

获取集合数据总量

zcard key

zcount key min max

集合交、并操作

zinterstore destination numkeys key [key ...] [AGGREGATE SUM|MIN|MAX]

zunionstore destination numkeys key [key ...]

获取珊瑚橘对应的索引（排名）

zrank key member

zrevrank key member

score值获取与修改

zscore key member

zincrby key increment member

score保存的数据存储空间是64位

score保存的数据也可以是一个双精度的double值，基于双精度浮点数的特征，可能会丢失精度，使用时要慎重；

sorted_set底层存储还是基于set结构的，因此数据不能重复，如果重复添加相同的数据，score值将被反复修改，保留最后一次修改的结果

获取当前系统时间

time

通用命令

key是一个字符串

针对key的操作

del key

exists key

type key 获取key的类型

为指定key设置有效期

expire key seconds

pexpire key milliseconds

expireat key timestamp

pexpireat key milliseconds-timestamp

获取key的有效时间

ttl key

pttl key

切换key从时效性转换为永久性

persist key

查询key

keys pattern

查询模式规则

1		*	匹配任意数量的任意符号	?	匹配一个任意符号
2		keys *	所有		
3		keys it*	查询所有以it开头		
4		keys *heima	查询所有以heima结尾		
5		keys ??heima	查询所有前面两个字符任意，后面以heima结尾		
6		keys user:?	查询所有以user:开头，最后一个字符任意		
7		keys u[st]er:1	查询所有以u开头，以er:1结尾，中间包含一个字母，s或t		

为key改名

rename key newkey

renamenx key newkey

对所有key排序

sort

其他key通用操作

help @generic

数据库通用操作

切换数据库

select index

其他操作

quit 退出

ping 检查服务器是否能联通，不写key的话能联通则返回PONG

echo message

数据移动

move key db

数据清除操作

dbsize

flushdb

flushall

Jedis